

**УДК 004.522**

*В. Ю. Герман, студент гр. ПГ-81мн, к.т.н., доц. О.М. Павловський*  
КПІ ім. Ігоря Сікорського

## **РЕАЛІЗАЦІЯ АЛГОРИТМУ КЕРУВАННЯ ПОЛЬОТНОГО КОНТРОЛЕРУ БПЛА ЗА ДОПОМОГОЮ SIMULINK**

**Анотація.** У роботі розглянуто сфери застосування БПЛА, переваги та недоліки методів реалізації створених користувачем алгоритмів керування на польотні контролери, методи роботи з контролером польоту Pixhawk, сфери його застосування та основні характеристики. Наведений алгоритм формування та завантаження технологічного програмного забезпечення для керування польотного контролеру, показана можливість опитування бортових навігаційних датчиків за допомогою Simulink. Визначено основні переваги такого підходу.

**Ключові слова:** БПЛА, алгоритми керування, польотний контролер, Pixhawk, Simulink.

### **ВСТУП**

Навігаційно-пілотажні системи сучасних безпілотних літальних апаратів (БПЛА) відрізняються своєю структурою, що змінюється в залежності від типу літального апарату, його маси, аеродинамічної схеми, вимог, що пред'являються до точності управління, стійкості та автономності. Але в цілому для коректної роботи БПЛА потрібно вирішити дві задачі: задачу навігації, яка полягає у визначенні координат літального апарату та його орієнтації в просторі, і задачу пілотування, яка полягає в управлінні літальним апаратом по навігаційним параметрам. На сьогоднішній день розроблена велика кількість контролерів польоту з власним програмним забезпеченням, які дозволяють вирішувати ці задачі. Наприклад: Multiwii, ArduCopter (APM 2.6, APM 2.8), Pixhawk, контролери DJI, MicroKopter, Zero UAV X4 / X6, AutoQuad, KK., XAircraft та ін [1-3]. Проте, в залежності від поставлених задач, часто виникає необхідність редагування алгоритмів, на яких працює польотний контролер, для підвищення їх точності, проведення досліджень, або створення власного режиму польоту. Проте більшість програмного забезпечення або немає можливості редагування коду, або його зміна вимагає додаткового спеціалізованого обладнання, або модифікація такого алгоритму потребує певної послідовності дії і програмного забезпечення, що не є тривіальним питанням. Таким чином, метою даної роботи є послідовний опис алгоритму зміни стандартного програмного забезпечення поширеного польотного контролеру Pixhawk.

### **ОГЛЯД РОБІТ, ЩО БУЛИ ПРОВЕДЕНІ ЗА СУМІЖНИМИ ТЕМАМИ**

В статті [4] представлено послідовність створення нового режиму польоту: ідентифікація нелінійної системи на основі нечіткої кластеризації та розробка контролера на основі Pixhawk для квадрокоптера. Для комунікації між контролером і комп'ютером використовувався пакет даних MAVROS який дозволяє дистанційно керувати БПЛА, вносити корективи і знімати покази з датчиків, але потребує глибокого знання мови C++ і комп'ютера з операційною системою Linux.

В роботі [5] були розглянуті різноманітні види контролерів з відкритим кодом, та побудований БПЛА на основі Pixhawk mini для отримання даних під час польотних випробувань і використання цих даних для перевірки і

вдосконалення конструкції БПЛА. Для знання даних і планування польоту у цій роботі використовувалася програма QGroundControl (QGC).

Існує доволі багато проектів, які можна об'єднати під однією назвою ArduCopter [2], суть проекту – використання у якості польотного контролеру платформи Arduino, що з одного боку спрощує створення програмної реалізації алгоритмів керування, а з іншого, за рахунок не оптимізованого коду – значно зменшує його функціональні можливості.

Методи реалізації створеного користувачем алгоритму керування на польотних контролерах БПЛА, наведені у вищезазначених роботах потребують знань для роботи з різноманітним програмним забезпеченням та навичок у програмуванні що робить ці методи досить складними у відтворенні. Таким чином, у подальшому будемо розглядати спрощені методи реалізації алгоритму керування та перевірки його на контролері польоту БПЛА.

### **РЕАЛІЗАЦІЯ МОДЕЛІ SIMULINK НА КОНТРОЛЕРІ PIXHAWK**

Для створення добре функціонуючого БПЛА, потрібно завчасно розроблений теоретичний алгоритм завантажити до польотного контролера, який в свою чергу, зв'яже всю апаратну складову БПЛА та керує нею. На цьому етапі виникає ряд проблем, наприклад, потрібно перевірити працездатність та перетворити алгоритм у код, який буде сумісний з контролером. Контролери польоту закритої системи, взагалі, не дозволяють будь яке внутрішнє втручання у код їх програми та перепрограмування ядра польотного контролера, тому для здійснення поставленої задачі необхідно використовувати польотні контролери, створені у рамках проектів з відкритою системою та кодом, такі як Pixhawk, Ardupilot, Paparazzi, Openpilot, та ін. Такі контролери постачають разом з комплектами розробки та редагування програмного забезпечення (Software Development Kit (SDK)), щоб користувачі могли розроблювати власні програми та режими польоту.

Для реалізації задачі був вибраний контролер Pixhawk, так як він є одним з найефективніших, з точки зору гнучкості системи, польотних контролерів доступних на цивільному ринку. Це система автопілоту побудована на базі проекту з відкритим кодом PX4 і вироблена компанією 3D Robotics. У більшості випадків, питання перенесення алгоритму на польотний контролер ускладняється тим, що оператор повинен володіти мовами програмування, що сумісні із наявним програмним забезпеченням, на досить високому рівні. Вирішення такої проблеми є використання пакету Simulink системи MatLab, що дозволить використовувати графічний спосіб створення алгоритму. Це середовище якнайкраще підходить для створення польотного алгоритму у вигляді блок-схеми, симуляції зовнішніх збурень та вхідних сигналів на систему, а також використовуючи додаткові налаштування, автоматично генерує готовий код для завантаження у контролер Pixhawk.

Проте, для реалізації зазначеного алгоритму, окрім пакету Simulink, необхідно мати наступний пакет програмного забезпечення: Simulink Coder, Embedded Coder, Aerospace Blockset, Pixhawk Toolchain та CMake. Докладна

інструкція по встановленню та налаштуванню приведеного вище програмного забезпечення викладена у роботі [6].

Після побудови моделі алгоритму польоту БПЛА у Simulink, розгортання створеної прошивки, користувачу необхідно натиснути на кнопку «Build», яка розташована на командній панелі інтерфейсу Simulink, після цього програмою виконується наступна послідовність дій:

- 1) Генерація коду моделі Simulink та компіляція в об'єктні файли;
- 2) Перенесення згенерованого коду у директорію \px4\Firmware\src\modules\px4\_simulink\_app разом з CMakefile.txt, який описує необхідні вихідні файли, включаючи шляхи та параметри компілятора;
- 3) Викликання CMake команд для генерації виконавчого коду контролера Pixhawk та інтеграції в неї коду з моделей Simulink;
- 4) Образ прошивки (\*px4. файл) розміщується у \px4\Firmware\build\_px4fmu-v2\_default;
- 5) Користувачеві буде запропоновано підключити контролер Pixhawk для завантаження готової прошивки.

Для перевірки підключення і коректної роботи контролера була створена тестова прошивка у вигляді блок-схеми на рис.1, для зняття значень з гіроскопу на протязі 10 секунд. На графіку з рис.2, зображений не оброблений сигнал гіроскопів контролера, який знаходився у стані спокою на горизонтальній площині.

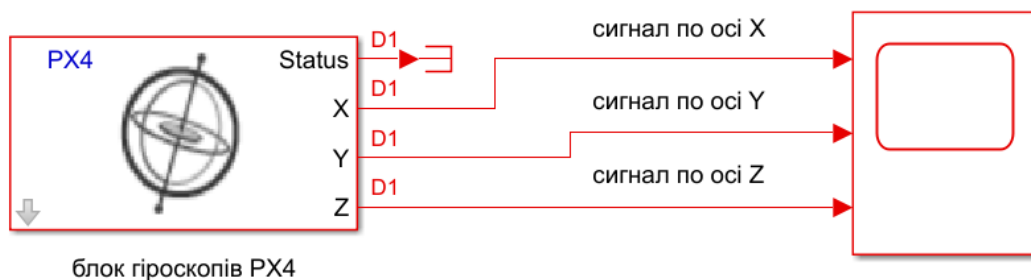


Рисунок 1. Simulink блок-схема виводу значень з трьох-вісного гіроскопу контролера Pixhawk на графік.

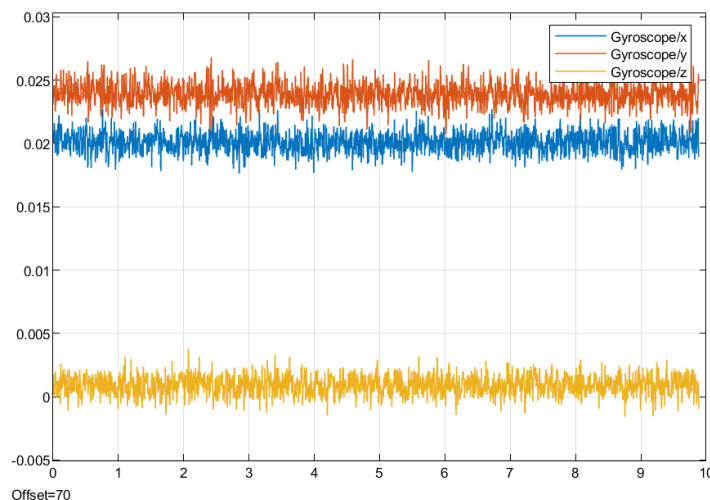


Рисунок 2. Покази з гіроскопу контролера за період в 10с.

Отримані дані можуть бути збережені у файл для подальшого аналізу на ПК, або для подальшої роботи можна модифікувати наведену блок-схему, додавши фільтри (Калмана, Мажвіка т.і.), що наявні у панелях інструментів Simulink. Такі дії призведуть до зменшення похибки і можуть бути використати для калібровки чутливих елементів, або для безпосереднього керування БПЛА. Таким чином, показана можливість простої реалізації будь-якого змодельованого алгоритму у Simulink на контролері Pixhawk, та можливість випробувати його на БПЛА, або вносити виправлення у наявний алгоритм керування.

## ВИСНОВКИ

У роботі приведені проекти, які реалізують БПЛА на готових польотних контролерах, розглянуті проблеми реалізації готової моделі керування на польотних контролерах різних виробників, зауважено на проблемах при адаптуванні власних алгоритмів до різних систем. Тому перевагу віддано польотним контролерам із відкритим кодом. Для перевірки працездатності та ефективності інтеграції розроблених алгоритмів був обраний польотний контролер Pixhawk і пакет Simulink системи MatLab. Створена тестова блок-схема, що реалізує зчитування значень із блоку гіроскопів, яка підтвердила простоту та ефективність обраного підходу. Також, не зважаючи на велику кількість додаткових програмних компонентів, такий підхід дозволяє досить легко корегувати алгоритм в моделі Simulink і відразу завантажувати його на польотний контролер для подальших випробувань на БПЛА.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Pixhawk | The hardware standard for open-source autopilots Guide [Електронний ресурс] – Режим доступу: <https://pixhawk.org/>
- [2] ArduPilot Documentation [Електронний ресурс] – Режим доступу: <https://ardupilot.org/ardupilot/>
- [3] MultiWii [Електронний ресурс] – Режим доступу: <http://www.multiwii.com/>
- [4] Sreenatha G., Matthew A. Fuzzy Clustering based Nonlinear System Identification and controller Development of Pixhawk based Quadcopter in International Conference on Advanced Computational Intelligence. – 2017. – № 9 – p. 223–230.
- [5] Shahla Pourkaram AIRCRAFT SYSTEM IDENTIFICATION USING THE PIXHAWK MINI CONTROLLER FOR ACADEMIC USE / Shahla Pourkaram // Diss. Wichita State University – 2018. – p.1–125.
- [6] Pixhawk Pilot Support Package (PSP) User Guide [Електронний ресурс] – Режим доступу: <http://discuss.px4.io/uploads/default/original/2X/d/d8a49f4c01c834a7d65472408731a80a57560356.pdf>

*Наук. керівник – к.т.н., доц. О.М. Павловський*